LibraHub

Los cuatro homies (Group 2)

Gabriel Arevalo, Daniel Gonzalez, Christopher Hicks, and Sukeer Ledezama

**Software Requirements Specification**

**Document**

**Version: (1.0)** **Date:(03/06/2024)**

Contents

[1. Introduction 3](#_Toc160644570)

[1.1 Purpose 3](#_Toc160644571)

[1.2 Scope 3](#_Toc160644572)

[1.3 Definitions, Acronyms, and Abbreviations. 3](#_Toc160644573)

[1.4 References 4](#_Toc160644574)

[1.5 Overview 4](#_Toc160644575)

[2. The Overall Description 5](#_Toc160644580)

[2.1 Product Perspective 5](#_Toc160644581)

[2.1.1 System Interfaces 6](#_Toc160644582)

[2.1.2 Interfaces 6](#_Toc160644583)

[2.1.3 Hardware Interfaces 6](#_Toc160644584)

[2.1.4 Software Interfaces 6](#_Toc160644585)

[2.1.5 Communications Interfaces 6](#_Toc160644586)

[2.1.6 Memory Constraints 7](#_Toc160644587)

[2.1.7 Operations 7](#_Toc160644588)

[2.1.8 Site Adaptation Requirements 7](#_Toc160644589)

[2.2 Product Functions 7](#_Toc160644590)

[2.3 User Characteristics 8](#_Toc160644591)

[2.4 Constraints 8](#_Toc160644592)

[2.5 Assumptions and Dependencies 8](#_Toc160644593)

[2.6 Apportioning of Requirements. 9](#_Toc160644594)

[3. Specific Requirements 10](#_Toc160644595)

[3.1 External Interfaces 10](#_Toc160644596)

[3.2 Functions 12](#_Toc160644600)

[3.2.1 Customer User Class 12](#_Toc160644601)

[3.2.2 Guest User Class 13](#_Toc160644602)

[3.2.3 Librarian User Class 13](#_Toc160644603)

[3.3 Performance Requirements 14](#_Toc160644604)

[3.4 Logical Database Requirements 16](#_Toc160644605)

[3.5 Design Constraints 17](#_Toc160644606)

[3.5.1 Standards Compliance 18](#_Toc160644607)

[3.6 Software System Attributes 18](#_Toc160644608)

[3.6.1 Reliability 18](#_Toc160644609)

[3.6.2 Availability 19](#_Toc160644610)

[3.6.3 Security 19](#_Toc160644611)

[3.6.4 Maintainability 20](#_Toc160644612)

[3.6.5 Portability 20](#_Toc160644613)

# 1. Introduction

## 1.1 Purpose

The purpose of this Software Requirements Specifications document is to provide customers and designers with what the project is about. It will tell them about what the project is about, interfaces, constraints, functions, and anything else that will be used in the library management system. The intended audience for this SRS will be for our stakeholders, designers, and software developers.

## 1.2 Scope

LibraHub is a software application aimed at modernizing library management by offering various functionalities. Users can sign up, search for media, check out books and materials, access checkout history, and place books on hold. The system automatically removes inactive users after a certain number of days to update the database to the current number of active users and free up Database capacity. Administrators can also add and remove books from the system. LibraHub simplifies the checkout process and allows librarians to update the catalog, enhancing the library experience.

## 1.3 Definitions, Acronyms, and Abbreviations.

**Words**

Cataloging. The process of organizing and describing library resources, such as books, journals, and multimedia, to facilitate efficient retrieval.

*Library System. A software application designed to manage and automate various library operations, including cataloging, circulation, and user management.*

*Media. A variety of different media like books, e-books, videos, articles, etc. A broad term to describe all these without listing them out.*

*Reservation System. A feature allowing users to reserve or place holds on library items that are currently checked out.*

*User. The person, or persons, who operate or interact directly with the product*

*User Authentication. The process of verifying the identity of library users, typically through login credentials, to ensure secure access to the system.*

**Acronyms and Abbreviations**

*API. Application Programming Interface*

*DBMS. Database Management System*

*ISBN. International Standard Book Number*

*GUI. Graphical User Interface*

*SRS. Software Requirement Specifications*

*UI. User Interface*

## 1.4 References

The application does not use any specific protocols, RFCs, etc., therefore, a reference is not needed.

## 1.5 Overview

The SRS will contain a description of the software including functions, interfaces, constraints, characteristics, assumptions, and requirements, and system attributes.

#### Section 1: Introduction

The first section of the SRS document introduces the project, outlining the application's purpose, project scope, and defining key terms and abbreviations. Additionally, it includes a list of references used in the development process.

#### Section 2: Overall Description

This section provides a comprehensive overview of the application. It covers the product perspective, functions, user characteristics, constraints, and assumptions related to the software application.

#### Section 3: Specific Requirements

The third section details the specific requirements for software development. It includes information on external interfaces, functions, performance requirements, logical database requirements, constraints, and software system attributes.

#### Section 4: Documentation Structure

# The final section is dedicated to making the SRS document user-friendly. It includes elements such as a table of contents for each section, an index, and appendices. Additional comments may be added at the end of this section if needed.

# 2. The Overall Description

## Product Perspective

The LibraHub Library Management system is a self-contained system. It will be catered to serve a local library.

A diagram of a library application

Description automatically generated

### 2.1.1 System Interfaces

1. The API being used for interface is JavaFX.
   1. Will be used for users to interact with the system application.
2. JDBC API for database connectivity.
   1. Will be used for the system to connect to the library’s database.

### 

### 2.1.2 Interfaces

1. GUI for users to interact with for the library management system.
2. No other interfaces will be needed since the user can use the GUI to interact with the application.
3. GUI interacts with the database to get any information, plus it interacts with the controller (code) to get information based off user inputs.
4. To optimize the interface:
   1. Prompts of directions for the user to follow.
   2. Buttons that are named on the GUI to make it readable for the user.
   3. Search bars that have prompt text

### 2.1.3 Hardware Interfaces

The system has no hardware interface requirements, since it will be run locally, therefore requiring only one type of hardware. The local machine will be a windows 10/11 computer that can be on the low-end of specs, and the application will still work.

### 2.1.4 Software Interfaces

8.0.36 MySQL Community Server. System must use MySQL server as its database component to make sure user and library information is stored on it. Communication with the database is through JDBC connections (remember to reference where JDBC is stated).

### 2.1.5 Communications Interfaces

The application is for a local library; therefore, it does not need web services since it will run on an application using a local database that is located in the library. Therefore, there is no need for local network protocols, deployment, etc.

### 2.1.6 Memory Constraints

There are no memory constraints to the program because the program is an application that can run on lower-end computers.

### 2.1.7 Operations

If the library shuts down their systems at midnight for data backup, it will not affect the design of the system. Since the system and the database are separate entities, if the database is being backed up, the system itself will not need to be changed/be impacted.

### 2.1.8 Site Adaptation Requirements

This application will not be deployed onto a site, therefore, there is no need for any site adaptation.

## 2.2 Product Functions

1. User authentication
   1. Each user will need an account to access the library management system. The best way to authenticate the user is through their unique username and password. Both attributes will be stored and saved in a database.
2. Borrowing and returning
   1. Once the user is logged into the library management system, if the user chooses to borrow a book, all they should do is look up a book and click a button that allows them to borrow it. The same applies for returning, the user would have to click a return button for returning the book to the library.
3. Book management
   1. Book management would be organized through metadata. Metadata could include information about the book such as the title, author, publication date, ISBN, genre, publisher and more. All this information would be stored in a database as well.
4. Notifications
   1. Notifications would provide a notice through the user about a book that’s overdue for returning, it could also serve as notifying the user of a book or media that is now available in the library. Notifications could also be used for fees.
5. Security
   1. The first form of security in the library would be user and administrator authentication. Secure coding will be vital in providing security in the application. For user information, when any data is collected, personal identifiers would have to be removed to protect user privacy.
6. Reservation
   1. Once a user picks out the sort of media they would like to borrow from the library, the media would be reserved for the user that requested it for a limited amount of time before the user has to return it.
7. Search and Retrieval
   1. If a certain media hasn’t been returned, the administrator would be able to access the user’s account only to retrieve the specific media. Since media will have a license of copyright, this limits the number of users who are able to borrow certain media. This makes it easier for the administrator to manage and retrieve media.
8. Reporting analytics(overdue media)
   1. Any sort of media that is overdue would be placed in a certain database that records and holds information about overdue media.

## 2.3 User Characteristics

The system caters to three main user categories. General users, with an educational level of 10 and above, typically possess minimal to no experience and exhibit little to no technical expertise. Admins and librarians, requiring an educational level of 18 and above, maintain minimal job-specific experience and possess only basic technical skills necessary for their roles. Lastly, guest users, sharing an educational level of 10 and above with the general users, similarly have little to no experience and minimal technical expertise when engaging with the system.

## 2.4 Constraints

The application does not have anything that will limit the developer’s options. There are no policies that will affect the application, no hardware limitations since the application can run on even lower-end computers.

## 2.5 Assumptions and Dependencies

The following factors are assumed to be true for the development of LibraHub and may impact the requirements stated in the SRS if the assumed factors are changed:

1. Operating System Availability: The SRS assumes that the designated hardware for LibraHub will have the required operating system available. Any changes in the availability or version of the operating system may require corresponding changes to the software requirements.
2. Internet Connectivity: The system relies on a stable internet connection only for connection to the database to retrieve information from it.
3. Hardware Compatibility: LibraHub is assumed to be compatible with computers. Changes in hardware specifications or additions of new hardware (like adding barcode scanners) may require adjustments to the software.
4. Library Policy Changes: The software assumes adherence to the existing library policies regarding user accounts, media checkout, and book management. Any changes in these policies may necessitate updates to the software to align with the new policies.
5. User Computer Literacy: The software assumes a basic level of computer literacy among users, including the ability to navigate the user interface, perform searches, and manage their accounts. Changes in user computer literacy may require adjustments to the software's user interface or help documentation to assist users with varying levels of proficiency.
6. Regulatory Changes: Changes in laws or regulations at the national, state, and municipal level relating to data privacy, or security may require updates to the software to ensure compliance and protect user data.
7. Third-Party Integration: LibraHub integrates with third-party services for certain features, such as eBook access. Changes in third-party services, including APIs, terms of service, or availability, may require updates to the software to maintain functionality.
8. Security Requirements: The software is expected to meet the security requirements specified by the library, including data encryption, access control, and protection against cyber threats. Any changes in security requirements may impact the design and functionality of the software.
9. Resource Availability: The development and maintenance of LibraHub rely on the availability of resources, including funding, personnel, and technology infrastructure. Any changes in the availability or allocation of these resources may impact the development timeline and software features.

## 2.6 Apportioning of Requirements.

All the requirements will be met by the deadline of May 1st, 2024. No features will be delayed until the future versions of the system.

# 3. Specific Requirements

## 3.1 External Interfaces

## Library Database Interface

1. Interaction with a database management system to store and retrieve information about books, users, transactions, etc.
2. Input: Data entered by librarians or users, such as book details, borrower information, and transactions.
3. Output: Displaying search results, borrower details, and other relevant information to library staff and users.
4. Valid range, accuracy and/or tolerance
   1. Book ISBN: the length and format of ISBN numbers.
   2. User IDs:  user identification numbers.
   3. Book Availability Status: real-time availability status updates.
   4. Due Dates:  calculating and displaying due dates for borrowed items.
   5. Fine Calculation: Specific tolerance for fine calculations to account for potential rounding errors.
   6. Data Entry Errors: Allow for a tolerance in data entry, considering potential typos or minor discrepancies.
5. Units of measure
   1. Quantity of Books: Unit of measure for the quantity of books, such as "copies" or "volumes."
   2. Time: U nit of measure for time, especially in relation to due dates and loan durations. Common units might include "days" for loan periods.
   3. Fines: Specific the unit of measure for fines, such as a currency for monetary penalties.
   4. User Data: Units for user-related data, such as "years" for age or "ID numbers" for user identification
6. Screen formats/Organization
   1. Search Interface: Input fields for searching books by title, author, ISBN, etc. Search results displaying relevant book information.
   2. Book Details: Detailed view for each book, showing information like title, author, publication date, and availability status.
   3. User Interface: User profiles with details such as name, contact information, and borrowing history.
   4. Transaction Screens: Screens for borrowing and returning books, including user and book identification.
   5. Administrative Interface: Screens for librarians or administrators to manage book inventory, user accounts, and generate reports.
   6. Notifications: Screens for displaying alerts or notifications, such as overdue book reminders.
   7. Dashboard: An overview screen summarizing key statistics, like available books, borrowed items, and fines.
7. Data Formats
   1. Books: Columns for ISBN, title, author, publication date, and availability status.
   2. Users: Columns for user ID, name, contact information, and borrowing history.
   3. Transactions: Columns for transaction ID, book ID, user ID, loan date, return date, etc.
   4. Categories/Genres: Columns for category/genre ID, name, and descriptions.
   5. Text/String: Used for information like book titles, author names, and user details.
   6. Numeric: Values such as ISBN numbers, user IDs, or quantities
   7. Date/Time: timestamps for transactions, due dates, and publication dates.
   8. Boolean: Representing binary states like book availability (true/false).
   9. Currency: System handles fines or monetary transactions.
8. Command Formats
   1. SQL Commands
   2. API Requests
   3. Management Commands
   4. Custom Commands
9. A closing or concluding message displayed on the user interface. This message might provide a summary, confirmation, or acknowledgment after a user completes a transaction, search, or other activities within the system. It ensures users receive clear feedback or closure for their interactions with the library database.

## Authentication Systems

## Integration with external authentication systems, such as LDAP or single sign-on (SSO), for user login and authentication.

1. Inputs
   1. Username, Password, One-Time Codes
2. Output
3. Authentication Success, Authentication Failure, Two-Factor Authentication Prompt, Account Lockout Warning
4. Valid Range:
   1. Username Length: Specifying a valid range for the length of usernames.
   2. Password Complexity: Defining requirements for the valid range of characters, length, and complexity for passwords.
   3. Password Retry Attempts: A tolerance level for the number of unsuccessful passwords attempts before implementing security measures like account lockout.
5. Units of Measure
   1. Time: Measure the duration of a session or the time taken for authentication processes.
   2. Attempts: Count the number of login attempts, considering it as a measure of user activity or system security.
6. Screen Formats/Organization
   1. Login Screen: Input fields for username and password. Option for two-factor authentication. "Forgot Password" link for password recovery.
   2. User Dashboard: After successful authentication, a dashboard displays relevant information. Options for updating security settings or managing account details.
   3. Error Messages: Clearly formatted error messages for unsuccessful authentication attempts. Instructions on resolving common issues.
7. Data Formats: relevant for storing and processing information such as:  
   User Data Format, Token Format, Logs Format, Error Messages Format, Session Data Format
8. Command Formats
   1. User Authentication Command, Token Generation Command, Password Reset Command
9. End Message
   1. Successful Authentication: Authentication successful.
   2. Unsuccessful Authentication: Invalid credentials.
   3. Account Lockout: Too many unsuccessful attempts.

## 3.2 Functions

There are three classes/objects that are used during our application. Those are the customer objects, guest object, and librarian object. Librarian and customer class have the same stimulus, which is logging in with their information. The guest class does not have a stimulus since it does not have an account, therefore, it does not need a stimulus.

### 3.2.1 Customer User Class

A customer is a user that has an account with the local library and can check out books, hold books, etc.

#### 3.2.1.1 User Authentication

The system shall check to make sure that the user has an account with the library by checking if their ID is in the system, then checking to see if their password is correct.

#### 3.2.1.2 Checking Account

The system shall allow users to check their accounts to see what holds, or media they currently have out by pulling up their information from the database.

#### 3.2.1.3 Overdue Media

The system shall check whether the user has overdue media, and if so, they will not be allowed to check out another book by checking whether the media that has been checked out is past the due date.

#### 3.2.1.4 Checking Out Books/Holding Books

The system shall allow users to check out books and hold books from the system, unless they have overdue media by checking their account.

#### 3.2.1.5 Checking Catalog

The system shall allow users to look up media within the library’s database.

### 3.2.2 Guest User Class

A guest is a user that does not have an account with the local library but can still browse the catalog.

#### 3.2.2.1 Checking Catalog

The system shall allow the guest to look up media within the library’s database.

#### 3.2.2.2 Checking/Holding Books

The system shall not allow the guest to be able to check out or hold books unless they have an account.

#### 3.2.2.3 Checking Account

The system shall not allow the guests to check their account because they do not have an account with the library.

#### 3.2.2.4 Creating Account

The System shall allow guests to create an account with the library by getting their information, then adding it to the library database.

### 3.2.3 Librarian User Class

A librarian is an administrator that can look up user accounts, check holds, add, or remove books, etc.

#### 3.2.3.1 Checking Accounts

The system shall allow a librarian to check user accounts to see their information by getting the user’s ID, and then getting the information from the database.

#### 3.2.3.2 Creating Accounts

The system shall allow a librarian to create an account for a user by getting their information, then adding it to the library database. The system shall also allow existing librarians to create a new librarian/administrator account for new librarians.

#### 3.2.3.4 Adding Media

The system shall allow a librarian to add media from their database. To add media, they would need the information about the media, then it would be added to the database.

##### 3.2.3.5 Removing Media

The system shall allow a librarian to remove media from their database by checking whether the media they want is part of the database, if so, they will receive confirmation if they want to remove it.

#### 3.2.3.6 Checking Catalog

The system shall allow a librarian to look at the catalog of books the library has.

#### 3.2.3.7 Librarian Authentication

The system shall check to make sure that the user is a librarian by checking if their ID is in the system, and checking whether they are an admin. Then the system also checks if they input the correct password that is matched with their ID.

#### 3.2.3.8 View Holds/Overdue Media

The system shall allow the librarian to check whether a user has hold or overdue media on their account.

## 3.3 Performance Requirements

The following indicates the Static and Dynamic numerical requirements of the LibraHub Library management system. Please keep in mind that the Dynamic and static numerical requirements reflect a local library user base that serves a population of approximately 5,000 people.

#### Static Numerical Requirements

* Number of simultaneous Users
  + The system should support at least 20 simultaneous users to ensure smooth operation during peak hours.
* Amount and Type of Information
  + The system should be able to handle a database of up to 5,000 books and 2,000 users, including information such as book titles, authors, ISBNs, user accounts, and borrowing histories.
* Library collection size
  + The system should be capable of managing a library collection of up to 50,000 items, including books, DVDs, CDs, and other media types.
* Book Checkout and Return
  + The system should be able to process at least 100 book checkouts and returns per hour during peak hours of operation.
* Book reservation
  + The system should support the reservation of at least 20 books simultaneously to accommodate user requests for popular items.
* User Account Management
  + The system should allow librarians to create and manage user accounts efficiently, with the ability to handle at least 50 new user registrations per day.
* Database Performance
  + The system should be able to retrieve and display search results for a typical query within 1-10 seconds to provide a responsive user experience.
* System Availability
  + The system should be available for use by library staff and patrons at least 99% of the time, with scheduled maintenance periods communicated in advance.

#### Dynamic Numerical Requirements

* Transaction Processing time
  + 90% of transactions, such as book checkouts and returns, should be processed in less than 5 seconds under normal workload conditions.
* Task completion time
  + The system should complete common tasks, such as searching for a book or adding a new user, within 3 seconds under normal workload conditions.
* User interaction Response time
  + The system should respond to user interactions, such as clicking on a button or entering text, within 0.5 seconds to provide a smooth and responsive user experience.
* Database Query performance
  + The system should be able to retrieve and display search results for a typical query within 1-2 seconds to ensure efficient access to information.
* Concurrency
  + The system should be able to handle at least 20 concurrent transactions, such as book checkouts or returns, during peak hours to prevent delays and ensure timely processing.
* Data storage capacity
  + The system should be able to store and manage data for at least 5 years, including user accounts, borrowing histories, and library collection information.
* Error handling
  + The system should handle errors gracefully, with no more than 3% of transactions resulting in system errors or crashes under normal operating conditions.

## 3.4 Logical Database Requirements

Entities

* User
  + Attributes
    - User ID (PK)
    - Name
    - Email
    - Password
    - Registration Date
    - Last Online
* Media
  + Attributes
    - Media ID (PK)
    - Title
    - Author
    - Type
    - Status
      * Available
      * Checked out
      * On hold
* Checkout
  + Attributes
    - Check Out ID (PK)
    - User ID (FK)
    - Media ID (FK)
    - Check Out Date
    - Return Date
* Hold
  + Attributes
    - Hold ID (PK)
    - User ID (FK)
    - Media ID (FK)
    - Hold Date
* Admin
  + Attributes
    - Admin ID (PK)
    - Name
    - Email
    - Password

Relationships

* One User can have multiple Checkouts and Holds (One-to-Many relationship).
* One Media can have multiple Checkouts and Holds (One-to-Many relationship).
* One Admin can manage multiple Media (One-to-Many relationship).

Additional Features

* User inactivity
  + Add an Inactivity Date attribute to the User entity to track the last activity date.
  + Implement a mechanism to periodically check for inactive users and remove them from the system.

## 3.5 Design Constraints

Design Constraints:

* Scalability – When developing the library management system, it should be designed to handle growth and change. Can the system potentially handle itself well when more media is added, or if more users register into the library management system?
* Performance – The system should be designed to have quick response times and should handle peak loads efficiently. This is vital because libraries tend to have a large user base and extensive digital collections.
* Accessibility – The library management system should be designed in a way where anyone can access it with ease. Those with a wide range of disabilities should also find using the application easy.
* User experience – The interface of the system should be easy to navigate, and meet the needs of the general audience, which can range from casual users, students, or researchers.
* Device compatibility – The system is designed to only be handled by computers; other forms of devices are not in mind for this application.
* Language and localization – In the case of this library management system, it’s designed to be local and to only handle the English language.
* Data Privacy and Security – security measures must be placed in the system to protect user data and privacy from any potential perpetrator.
* Organization – all sorts of media like books, movies and more will have to be organized through specific attributes and identifiers in the database they belong to.

### 3.5.1 Standards Compliance

A regulation will be following data protection regulations and standards, to ensure confidentiality and privacy of user information. Furthermore, the system must also comply with data protection laws, and other local regulations,

## 3.6 Software System Attributes

Software system attributes refer to qualities or characteristics that define the behavior and performance of the system.

* Reliability: Ensure the library software functions correctly and consistently under various conditions
* Availability: The system should be accessible when needed, minimizing downtime.
* Performance: Define acceptable response times and efficiency benchmarks for tasks like searching, borrowing, and returning books.
* Scalability: Consider the ability to handle a growing number of users, books, and transactions.
* Security: Implement measures to protect user data, transactions, and system integrity.
* Maintainability: Design the system for easy updates, bug fixes, and enhancements to ensure long-term viability.
* Usability: Prioritize a user-friendly interface and intuitive workflows for both librarians and patrons.
* Interoperability: Ensure compatibility with existing library systems or external databases for seamless integration.
* Portability: Design software to run on different platforms or environments without significant modifications.
* Compliance: Adhere to relevant standards, regulations, and industry best practices.

### 3.6.1 Reliability

For the system to fully work on the time of delivery, will be for the system to be able to connect to the library’s current database. This way any information from the old system can be ported over to the new system (i.e. user data, catalog, etc.) There is no meantime between failures.

### 3.6.2 Availability

Media availability varies, with some items having multiple copies and others just one. Users check availability for borrowing; for example, if all copies of a book are borrowed, the borrow button is inactive. The system's catalog facilitates easy media lookup. The local library system operates during set daily hours (e.g., 8:00 am to 5:00 pm) to optimize resource usage compared to a 24/7 system.

### 3.6.3 Security

The following Factors will be implemented into the LibraHub software to protect the software from accidental or malicious access:

* Keeping a log history of user activities
  + The system should maintain detailed logs of user activities, system events, and data access to facilitate auditing and forensic analysis in case of security incidents.
* Access Control
  + The system will Implement strict access control mechanisms to ensure that only authorized users have access to the system and its data. This can include using strong passwords, multi-factor authentication, and role-based access controls.
* Data Backup
  + Regularly back up data to protect against data loss due to accidental deletion, hardware failure, or malicious attacks. Store backups securely and test them regularly to ensure they can be restored when needed.
* User Education
  + Educate users about best practices for security, such as choosing strong passwords, recognizing phishing attempts, and reporting suspicious activities.
* Secure Development practices
  + The development team will follow secure coding practices and conduct regular security reviews and penetration testing to identify and mitigate potential vulnerabilities in the software.
* Prepared Statements
  + Use parameterized queries or prepared statements to construct SQL queries. This approach separates SQL code from user input, making it impossible for attackers to inject malicious SQL code.
* Database Permissions
  + Use the principle of least privilege to assign permissions to database users. Avoid granting excessive privileges that could allow attackers to execute arbitrary SQL commands.

### 3.6.4 Maintainability

The software will be broken down by essentially “pages” for easier maintenance. This way, if a page breaks down, or something needs to be updated, we can go to the specific page, troubleshoot, and fix it without messing up the other pages of the application. Furthermore, we will have comments within the code for easier understanding of the application whenever it needs to be maintained.

### 3.6.5 Portability

The code of the software and its components are entirely host-dependent, meaning that it can run on any computer. The only thing that would need to be changed would be the connection to the database that is “if” the system was to be ported to other libraries, which it is not. Furthermore, the application will be an .exe file, allowing you to run on Windows operating systems.